

Amendments to the Claims

Kindly amend claims 1, 11, 21 & 22, and cancel claims 10, 20 & 31 (without prejudice) as set forth below. In accordance with current amendment practice, changes in the amended claims are shown by underlining (for added matter) and strikethrough (for deleted matter).

1. (Currently Amended) A method of communicating between programs having different machine context organizations, said method comprising:

determining, at compile time, which savearea layout of a plurality of savearea layouts is to be used to save information relating to a calling program; and

selecting, at compile time, a linkage service from a plurality of linkage services to be used in communicating between said calling program and a callee program, wherein the linkage service comprises an intermediary service routine between the calling program and the callee program, wherein said selecting is based upon the determined savearea layout, and wherein said calling program and said callee program coexist within a single executable module but have different machine context organizations, the different machine context organizations comprising different register widths.

2. (Original) The method of claim 1, wherein said determining is based upon one or more attributes of said callee program.

3. (Original) The method of claim 2, wherein one attribute of said one or more attributes comprises a size of one or more registers to be used by said callee program.

4. (Original) The method of claim 2, wherein said determining is further based on a target architecture mode.

END920000013US1

-2-

5. (Canceled).

6. (Original) The method of claim 1, wherein said linkage service comprises at least one of a calling service and a returning service.

7. (Original) The method of claim 1, wherein at least two savearea layouts of said plurality of savearea layouts coexist within a single executable module.

8. (Original) The method of claim 1, wherein said determining and said selecting enable use of a source code that has at least one of the following: a reduced amount of dual path source code, natural parameter passing to/from a variety of caller types, and natural exploitation of a large architecture, where desired.

9. (Original) The method of claim 8, wherein said source code comprises at least one common name usable in referencing one or more analogous fields in at least two savearea layouts of said plurality of savearea layouts to reduce dual path source code.

10. (Canceled).

11. (Currently Amended) A system of communicating between programs having different machine context organizations, said system comprising:

means for determining, at compile time, which savearea layout of a plurality of savearea layouts is to be used to save information relating to a calling program;
and

means for selecting, at compile time, a linkage service from a plurality of linkage services to be used in communicating between said calling program and a callee program, wherein the linkage service comprises an intermediary service routine

END920000013US1

-3-

between the calling program and the callee program, wherein said selecting is based upon the determined savearea layout, and wherein said calling program and said callee program coexist within a single executable module but have different machine context organizations, the different machine context organizations comprising different register widths.

12. (Original) The system of claim 11, wherein said means for determining comprises using one or more attributes of said callee program.

13. (Original) The system of claim 12, wherein one attribute of said one or more attributes comprises a size of one or more registers to be used by said callee program.

14. (Original) The system of claim 12, wherein the determining is based on a target architecture mode.

15. (Canceled).

16. (Original) The system of claim 11, wherein said linkage service comprises at least one of a calling service and a returning service.

17. (Original) The system of claim 11, wherein at least two savearea layouts of said plurality of savearea layouts coexist within a single executable module.

18. (Original) The system of claim 11, wherein said means for determining and said means for selecting enable use of a source code that has at least one of the following: a reduced amount of dual path source code, natural parameter passing to/from a variety of caller types, and natural exploitation of a large architecture, where desired.

19. (Original) The system of claim 18, wherein said source code comprises at least one common name usable in referencing one or more analogous fields in at least two savearea layouts of said plurality of savearea layouts to reduce dual path source code.

20. (Canceled).

21. (Currently Amended) A system of communicating between programs having different machine context organizations, said system comprising:

a computing environment adapted to determine, at compile time, which savearea layout of a plurality of savearea layouts is to be used to save information relating to a calling program; and

said computing environment being further adapted to select, at compile time, a linkage service from a plurality of linkage services to be used in communicating between said calling program and a callee program, wherein the linkage service comprises an intermediary service routine between the calling program and the callee program, wherein said selection is based upon the determined savearea layout, and wherein said calling program and said callee program coexist within a single executable module but have different machine context organizations, the different machine context organizations comprising different register widths.

22. (Currently Amended) At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method of communicating between programs having different machine context organizations, said method comprising:

determining, at compile time, which savearea layout of a plurality of savearea layouts is to be used to save information relating to a calling program; and

selecting, at compile time, a linkage service from a plurality of linkage services to be used in communicating between said calling program and a callee program, wherein the linkage service comprises an intermediary service routine between the calling program and the callee program, wherein said selecting is based upon the determined savearea layout, and wherein said calling program and said callee program coexist within a single executable module but have different machine context organizations, the different machine context organizations comprising different register widths.

23. (Original) The at least one program storage device of claim 22, wherein said determining is based upon one or more attributes of said callee program.

24. (Original) The at least one program storage device of claim 23, wherein one attribute of said one or more attributes comprises a size of one or more registers to be used by said callee program.

25. (Original) The at least one program storage device of claim 23, wherein said determining is further based on a target architecture mode.

26. (Canceled).

27. (Original) The at least one program storage device of claim 22, wherein said linkage service comprises at least one of a calling service and a returning service.

28. (Original) The at least one program storage device of claim 22, wherein at least two savearea layouts of said plurality of savearea layouts coexist within a single executable module.

29. (Original) The at least one program storage device of claim 22, wherein said determining and said selecting enable use of a source code that has at least one of the

END920000013US1

-6-

following: a reduced amount of dual path source code, natural parameter passing to/from a variety of caller types, and natural exploitation of a large architecture, where desired.

30. (Original) The at least one program storage device of claim 29, wherein said source code comprises at least one common name usable in referencing one or more analogous fields in at least two savearea layouts of said plurality of savearea layouts to reduce dual path source code.

31. (Canceled).

32. (Previously Presented) The method of claim 1, wherein said determining is based upon one or more attributes of said calling program.

33. (Previously Presented) The method of claim 1, further comprising performing said determining and said selecting for a caller program and a calling program having similar machine context organizations.

34. (Previously Presented) The method of claim 1, wherein said plurality of savearea layouts coexist within a single executable module, and wherein at least one savearea layout of said plurality of savearea layouts is usable when said calling program and said callee program have different machine context organizations and wherein at least one other savearea layout of said plurality of savearea layouts is usable when a calling program and a callee program have similar machine context organizations.

35. (Previously Presented) The system of claim 11, wherein said means for determining comprises using one or more attributes of said calling program.

36. (Previously Presented) The system of claim 11, further comprising means for performing said determining and said selecting for a caller program and a calling program having similar machine context organizations.

END920000013US1

-7-

37. (Previously Presented) The system of claim 11, wherein said plurality of savearea layouts coexist within a single executable module, and wherein at least one savearea layout of said plurality of savearea layouts is usable when said calling program and said callee program have different machine context organizations and wherein at least one other savearea layout of said plurality of savearea layouts is usable when a calling program and a callee program have similar machine context organizations.

38. (Previously Presented) The at least one program storage device of claim 22, wherein said determining is based upon one or more attributes of said calling program.

39. (Previously Presented) The at least one program storage device of claim 22, wherein said method further comprises performing said determining and said selecting for a caller program and a calling program having similar machine context organizations.

40. (Previously Presented) The at least one program storage device of claim 22, wherein said plurality of savearea layouts coexist within a single executable module, and wherein at least one savearea layout of said plurality of savearea layouts is usable when said calling program and said callee program have different machine context organizations and wherein at least one other savearea layout of said plurality of savearea layouts is usable when a calling program and a callee program have similar machine context organizations.